

**Listing of Claims:**

1-3 (canceled)

4. (currently amended) A computer-implemented document-searching method for searching a document having a hierarchical structure with elements separated by element identifiers, comprising the steps of:

5 generating an XPATH query automaton by storing a query expression input by a compiling device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of:

generating and registering a state transition by:

10 replacing an XPATH axis including an XPATH axis in a forward direction that is exemplified as an axis child, or a descendant in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in the an opposite direction that is exemplified as axis parent, ancestor in an XPath into a state transition,

15 replacing an XPATH axis including an XPATH axis in a direction of a following-sibling, or a preceding-sibling in said XPath into a state transition,

replacement of a predicate of an XPath into a state transition,

20 replacement of a logical product (and) of a predicate of an XPath into a state transition,

replacement of a logical add (or) of a predicate of an XPath into a state transition,

replacement of a logical NOT (not) of a predicate of an XPath into a state transition,

25                                   and a logical expression including a conjunction or a  
negative expression, while keeping an input query expression equal in terms of  
search, the query automaton including a plurality of states of the backward  
nodes, a condition for transition, and at least a search state, wherein said  
search state includes two states of said input query expression concurrently in a  
30 state transition, and wherein every axis regarding sibling relationship among  
nodes can be included in the search condition for said query automaton;

                                  said a query automaton evaluator determining a state transition of  
a node under determination by storing a left node and a lower node in  
correspondence with an identified element identifier, wherein the information  
35 obtained from said left node and information obtained from said lower node for  
a state transition is used concurrently[.]], and evaluating said query automaton  
with a search result of said left node and said lower node;

                                  storing the XPATH query automaton generated by said compiling  
device in a query automaton storage device; and

40                                   reading out said XPATH query automaton from said query  
automaton storage device and storing said query automaton, while reading in  
said document and performing a stream search with said a query automaton  
evaluator by using states of a plurality of different types of nodes in said  
element identifiers included in said document and said query automaton,  
45 thereby using two inputs and a search state[. ]; and

                                  storing the output of the query automaton evaluator in a ~~search~~  
~~result-storage device~~ means for, and thereafter outputting the stored output of  
the query automaton evaluator and the output of the searched node.

5.       (cancelled)

6.       (previously presented) The document-searching method according to  
Claim 4, wherein said step of generating an XPATH query automaton comprises

a step of generating an XPATH query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon.

7-13 (canceled)

14. (currently amended) A computer-implemented compiling method for generating a query automaton for performing a document search, comprising the steps of:

generating and registering a state transition by:

5                    replacing an XPATH axis including an XPATH axis in a forward direction that is exemplified as an axis child, or descendant in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in the an opposite direction that is exemplified as an axis parent, or ancestor in an  
10   XPath into a state transition,

replacing an XPATH axis including an XPATH axis in a direction of a following-sibling or a preceding-sibling sibling in an XPath into a state transition,

15                   replacement of a predicate of an XPath into a state transition,

replacement of a logical product (and) of a predicate of an XPath into a state transition,

replacement of a logical add (or) of a predicate of an XPath into a state transition,

20                   replacement of a logical NOT (not) of a predicate of an XPath into a state transition,,

and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of

search, and storing a plurality of states of a backward node in correspondence  
25 with said backward node into a query automaton storage device;

generating a query automaton by registering a plurality of states of  
said backward node, a condition for transition, at least a search state, wherein  
said search state includes two states of said input query expression  
concurrently in a state transition, wherein every axis regarding sibling  
30 relationship among nodes can be included in the search condition for said query  
automaton, and a reached state in correspondence with each other in said  
storage device, performing parsing, and identifying different types of nodes in  
said element identifiers, by the steps of

generating and registering a state transition by the query  
35 automaton including a plurality of states of the backward nodes, a condition for  
transition, and at least a search state,

said a query automaton evaluator determining a state  
transition of a node under determination by storing a left node and a lower node  
in correspondence with an identified element identifier, wherein the information  
40 obtained from said left node and information obtained from said lower node for  
a state transition is used concurrently, and

evaluating said query automaton with a search result of  
said left node and said lower node and wherein node data stored until then is  
cleared after said evaluating; and

45 storing the output of the query automaton evaluator in a ~~search~~  
~~result-storage device means,~~ and thereafter outputting the stored output of the  
query automaton evaluator and the output of the searched node.

15. (Original) The compiling method according to Claim 14, wherein said  
50 compiling method comprises a step of identifying said backward node as a left  
node or a lower node according to a type of said element identifier, and wherein  
said plurality of states are states of said left node and said lower node.

Appl. No. 10/670,068  
Amdt. dated July 17, 2008  
Reply to Office action of May 28, 2008

16-22 (canceled).